93-7

Squire, Sanders & Dempsey

Counsellors at Law
1201 Pennsylvania Avenue, N.W.
P.O. Box 407
Washington, D.C. 20044-0407

U.S. Offices:
Cleveland, Ohio
Columbus, Ohio
Jacksonville, Florida
Miami, Florida
New York, New York
Phoenix, Arizona

International Offices:
Brussels, Belgium
Budapest, Hungary
London, England
Prague, Czech Republic

Telephone: (202) 626-6600
Cable Squire DC
Telecopier: (202) 626-6780

Direct Dial Number

(202) 626-6634

July 6, 1995

BY HAND

DOCKET FILE COPY ORIGINAL

RECEIVED
JUL 6 1995
FCC MAIL ROOM

John T. Nakahata
Office of Chairman Reed E. Hundt
Federal Communications Commission
1919 M Street, N.W., Room 814
Washington, D.C. 20554

Re: Decoder Interface -- Echelon Corporation

Dear John:

As we discussed earlier today, enclosed is a copy of the report prepared by Diablo Research Corporation regarding the draft IS-105 Decoder Interface standard. As I explained, the report concludes that IS-105 will not favor any one home automation system, either economically or technically. The report also concludes that the proposed Decoder Interface standard "appear[s] to be reasonable and appropriate to meeting the goal of supporting features which the American consumer has come to expect from common entertainment electronics -- feature which are interfered with by traditional cable decoders."

After you have had an opportunity to review the enclosed report, please let us know if you have any questions.

Sincerely,

Joseph P. Markoski

Enclosure
cc: William F. Caton (for inclusion in the records of ET Docket No. 93-7)

No. of Copies rec'd 2
List A B C D E

Analysis of Draft EIA IS-105 Decoder Interface Standard

Competitive Effects on Home Automation Systems

Prepared for
Electronic Industries Association

by
Diablo Research Corporation
Sunnyvale, California

RECEIVED
JUL 6 1995
FCC MAIL ROOM

Abstract

Diablo Research has analyzed the draft proposal IS-105, Decoder Interface Standard, with emphasis on the potential for interfaces to home automation systems. While the Decoder Interface does make use of a subset of the CEBus* protocol to carry coordinating messages between devices, we find that the messaging format does not provide CEBus home automation manufacturers with a significant cost advantage over other manufacturers in offering an interface device to this bus.

This is true primarily because the proposal does not implement an object-oriented CEBus interface model for abstract user interface transactions. Specific functions dealing with the Decoder Interface problem do use an object-oriented structure since the task is well specified, but extended, unspecified services (such as home automation interfaces) simply use the messaging system to deliver limited transaction tokens between devices. A home automation interface will have to translate these tokens into its native protocol, regardless of which protocol that is. This requirement is no different for CEBus systems.

We further conclude that the physical hardware and functional requirements of the proposal appear to be reasonable and appropriate to meeting the goal of supporting features which the American consumer has come to expect from common entertainment electronics--features which are interfered with by traditional cable decoders.

Summary

The connection between a home automation (HA) system and the Decoder Interface can be made in two obvious ways. The first of these is to create a stand-alone interface device, separate from the other Receiver and Decoder devices, which serves to translate messages between the interface bus devices and the HA system. The primary function of this device would be to accept user input from the remote control, place information on the TV screen, and take appropriate action in the HA system (such as close drapes or open a garage door).

The second approach is to build HA system intelligence into the TV, since it is the natural user interface, and then go directly from the TV to the HA power line bus, thus skipping the Decoder Interface entirely. (It is a common assumption that the power line will be the workhorse communications medium for HA systems.)

We believe that the likely approach will be the stand-alone interface model. This is because the economics of producing TVs argue against attempting to support open-ended systems (such as HA systems) with processing in the TV. The features which the consumer will embrace and demand are not yet clear, and in addition, the electronic interfaces to the myriad of other home products have been and will be years in development, as the technology is refined and the market grows. Given this environment, the risk that a given implementation in a TV will fall short is high. Furthermore, the consumer's investment in the core function of this product is so high that replacement is not a reasonable option.

The stand-alone interface model, on the other hand, permits the HA functions to be placed in a small device which can be more easily upgraded or replaced as the HA marketplace matures. It also takes the cost of this interface out of the TV, which must generally be designed to meet the "lowest common denominator" to remain competitive.

To summarize, the argument against putting the HA intelligence in the TV also means that the TV cannot produce arbitrary HA messages.[1] The only rational solution to this problem is to define a generic set of tokens which the remote control generates and the TV simply forwards to the HA interface. The interface device interprets these commands in the context of the menu which it is presently generating on the screen. This approach is arbitrarily extensible to any functions, including but not limited to HA applications, and it requires no upgrades to the TV or the remote control device as new features are added.

This is the approach which the IS-105 developers have taken. The HA interface device would be a Decoder in IS-105 parlance; it can receive User input tokens (from the User's remote control), and it can ask the TV to put menus on the screen, either by sending characters to the TV's internal character generator[2], or by developing a video signal directly and sending that to the TV. The device then sends the necessary messages on the HA system to deal with the User's requests, such as to change the setpoint on the thermostat or recall the security system status. The TV never sends the actual CEBus message for changing the thermostat, and it never receives the CEBus security system status message. The interface device sends and receives the HA messages, and whether the HA system is CEBus compliant, or based on some other system such as LonWorks® is irrelevant—the task is equivalent.

Introduction

The goal of supporting features such as Picture-in-Picture displays, the recording of one channel while watching another, and the automatic (pre-programmed) recording of a given channel, sets some minimum requirements for the design of the solution. The additional goals of minimizing the total amount of circuitry (such as eliminating redundant tuners) to reduce cost, and minimizing the number of demodulation and remodulation steps (both to reduce cost and to avoid degrading picture and sound quality) also serve to constrain the design of the solution. We believe that the Decoder Interface proposal addresses these goals in a cost effective manner, by using a baseband signaling interface.

[1] The exception here would be if the remote control itself sends fully formatted HA messages. In this case the TV could still simply relay the message, and yet the Interface bus would carry true HA messaging. In order for this approach to work the remote control must be bi-directional, and it must be responsible for generating the screens. This is highly impractical in the near future, and we dismiss it for this reason. A practical, cost-competitive system cannot be based upon this approach at present

In order to achieve simultaneous use of two channels, baseband signaling requires multiple physical wires. In order to coordinate the use of these wires, and to support the sharing of a single tuner between the TV and the Cable box, for example, a communications scheme must be developed. Consumer electronics manufacturers have been developing a solution to a similar problem for years, under the name Audio Video (A/V) bus. Its goals included simplifying the wiring between VCRs, TVs and Stereo audio systems, and reducing the cost of moving high quality video from the VCR or Laser Disc Player to the TV (or to another VCR).

The size of the A/V bus is limited to a small "cluster" of entertainment devices, and the number of wires is limited to that needed to support two simultaneous audio and video sessions. Additional capability requires an additional bus. This was thought to be adequate for most scenarios, and adding more wires would have increased costs for all Users.

It is apparent to us that the application of the A/V bus design to the decoder interface problem is logical and reasonable. The decoder interface problem is very similar to the problems which the A/V bus was designed to address. The requirement to share tuners has resulted in the addition of the IF line.

The use of some of the CEBus protocol also appears to be a natural application of the A/V bus technology. To use or invent something else would have been to discard the result of years of work, and then develop another solution to the same set of problems. However, by recognizing the need to limit the TV's sophistication to a generic set of Decoder messages, IS-105 frees the TV from the need to understand all future HA and other messages, and puts all HA systems on equal footing in their use of the bus.

By means of comparison, the true CEBus video distribution system is a more versatile and more expensive system. It provides for 16 channels of simultaneous usage, rather than two, and covers the entire house rather than a small cluster. To support this approach the consumer must install a centralized RF processing device, the so-called Coax Node Zero, and all signal sources (such as VCRs, security cameras, etc.) must include frequency-agile modulators (fixed frequency possible but discouraged). The A/V bus was designed to be a limited, low cost alternative to a complete home automation system approach.

Decoder Interface Review

The proposed Decoder Interface is described in EIA IS-105.1, Decoder Interface Standard, and IS-105.2, Decoder Interface Control Standard. These documents have been developed by an EIA/NCTA Joint Engineering Committee, and only a brief review of this work will be presented here. The reader is urged to obtain these documents for a complete description. (See References section at end of this document.)

The Decoder Interface uses a 20 pin, "multi-pin connection" cable organized into nine differential twisted pairs, and two ground reference lines. The 9 pairs are used as follows: Four audio lines, four video lines, and a control message line which carries the control bus signals.

In addition, the Interface uses a single coax line to carry the down-converted, Intermediate Frequency (IF) signal from the Receiver to the Decoder. The Decoder uses this same wire to send a DC voltage back to the Receiver, to adjust the gain of the Receiver's tuner.

Normally the Receiver uses its own tuner circuitry to develop the audio and video signals which are presented to the User. In this case the Decoder box is essentially inactive, since its output signals are not being used. When a scrambled channel is tuned, however, the TV is not able to recover the video and audio—it must rely on the Decoder to do this.

In order to reduce cost and wiring complexity the decoder does not have a tuner. It uses the Receiver's IF signal as the input to its descrambling circuitry. The recovered audio and video signals are placed on several of the multi-pin connector's twisted pairs, and the Receiver uses these to present the channel to the User. This basic usage is depicted in Figure 1, below.

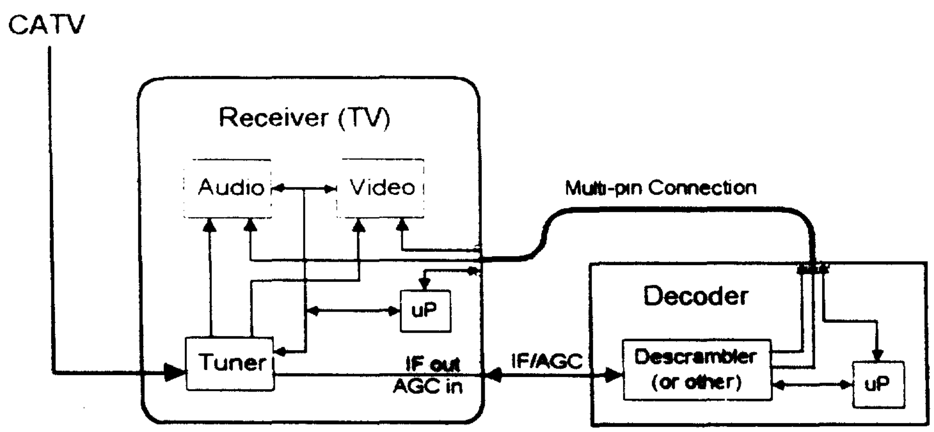

Figure 1. Basic use of Decoder Interface for Scrambled Channel

Since multiple Receivers and Decoders can share the interface bus, the actual details are more complex than this simple explanation suggests. The control bus is used to send messages which negotiate for the use of the shared lines. The control bus is also used to install new devices on the bus and to recover from any errors which may be detected.

The devices on the bus must maintain a standard set of variables which may be read by other devices on the bus. This permits the devices to be identified to the User with menus in plain English, etc., and permits other devices to learn some of the basic capabilities of each device. This in turn permits them to automate their interactions to a great extent, and thus reduce the effort of the consumer in configuring the system properly during set-up.

Other defined messages support features such as Receiver Monitoring, Channel Mapping, Data Channel Locking, and Direct User Interaction with Decoders. The first three features are among those which help to coordinate the Receiver/Decoder pair as the User moves from channel to channel. The last feature supports standardized support for undefined Decoder features. It is this feature which the HA interface device would use to bring the HA system to the TV screen and remote control keypad.

Home Automation Interface to IS-105

The following sections will present the HA Interface device as a Decoder device under IS-105, for the purpose of exploring the circuitry blocks which are required.

Before examining that approach we first present the logical alternative, as described in the Summary, above. Refer to Figure 2, below. In this approach the TV contains the HA interface circuitry and logic. For reasons stated previously we believe that this is the wrong approach. However, it should be clear that there are no inherent economic advantages for CEBus over other protocols in this approach, since no HA messaging is delivered on the IS-105 bus. (Whether IS-105 should use a CEBus subset at all is a different argument, as all implementations will share this cost equally.)

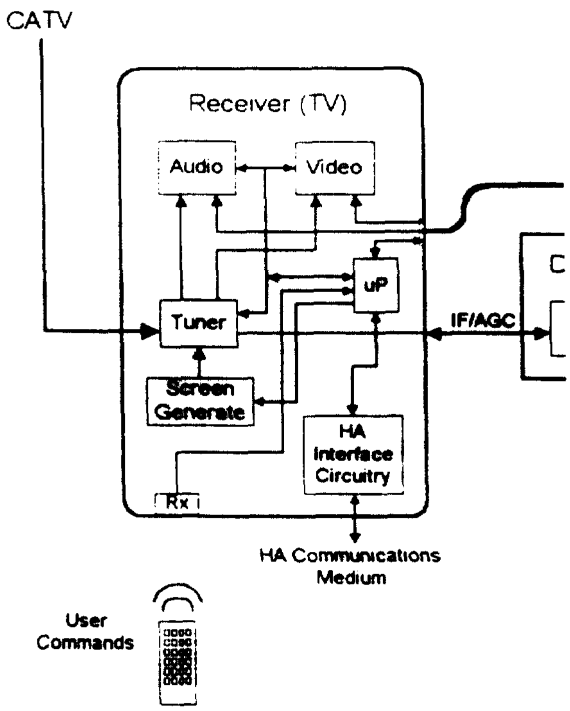

Figure 2. TV as Home Automation System Interface

We should note that Figure 2 is shown at a high level of abstraction. There are additional processors used in the communications interfaces, for example. The following Figures will move toward more detail, as we pursue the most likely architecture for HA Interface devices.

Figure 3, below, depicts the basic Receiver/Decoder pair, with a second Decoder serving as the HA Interface device. The HA Interface registers itself with the other devices on the bus, so that they can coordinate their interactions. The User must be supplied with a remote control which can generate a standard set of Decoder tokens. In addition, the remote must be able to select a particular decoder to which the tokens are to be addressed. With these features standardized, the User may direct commands to the HA Interface device when any of its features are to be accessed.

When the user directs a token to the HA Interface, it in turn takes action to cause the TV screen to come under its control.

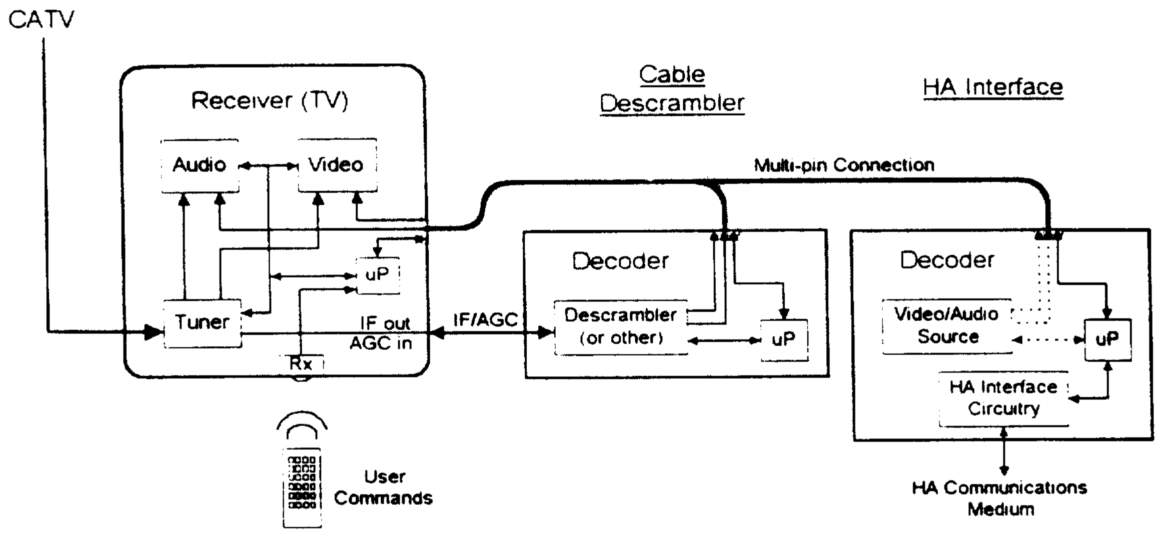

Figure 3. IS-105 Decoder as Home Automation System Interface

Once IS-105 is complete there may be a standardized way to use the character generator of the Receiver to place menus on the screen[2]. In the generalized case the Interface box will contain a video generator of its own, and potentially an audio generator which would allow voice prompts and sound effects to accompany the information display and user actions. In this case, when the user sends an initial token to the HA Interface the Interface must obtain the use of some of the lines on the bus, and must direct the TV to use these lines as the source of information for the screen/speakers.

Some of the bus lines may already be in use. For example, the VCR (another Receiver on the bus) may be recording a program which is being received by the satellite receiver (another Decoder on the bus). We would not want to interrupt this session by taking over these lines. IS-105 addresses this issue by providing a "Data Channel Locking" feature, which permits devices to guard lines which should not be interrupted. However, if the User were watching a scrambled program from the Cable Decoder, and then wanted to switch to the HA Interface for a time, the Cable box will relinquish its lines for that purpose as they need not be maintained while the User is not watching that source. Since these allocations are managed automatically they should not interfere with the User's perception of the equipment behavior.

[2] The HA Interface may be able to access the character generator of the TV, such that the Interface device sends text characters to the TV, and the TV places these characters on screen. In this case the Interface device need not contain any video generation circuitry, and need not gain access to IS-105 bus lines to carry the video to the TV. This will be a practical alternative only if all receivers are required to support this capability, so that a decoder manufacturer is guaranteed that this facility will be present.

Interface Device Block Diagrams

This section will examine the design of the IS-105 HA Interface Decoder in more detail, with emphasis on those features which will differ in two implementations as examples—an Interface to CEBus power line, versus an Interface to one of the LonWorks power line versions.

Figure 4, below, depicts one possible generic IS-105 HA Interface. The shaded blocks indicate the sections which are variable across different home automation systems. This model will be used for both CEBus and LonWorks interfaces, to evaluate obvious differences in equipment costs.

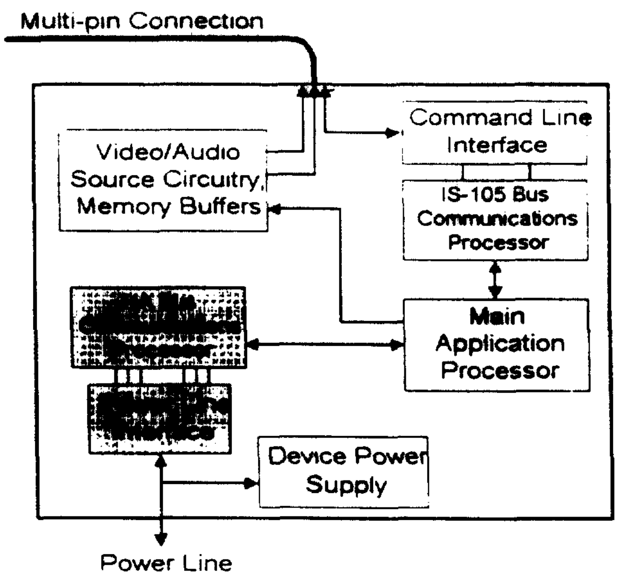

Figure 4. A Generic IS-105 Home Automation System Interface

Before moving on to HA-protocol-specific examples, let us review those circuit blocks of Figure 4 which will be common to any implementation. Figure 5 presents a more detailed diagram of the circuit elements which are unchanged by the selection of HA protocol. The following paragraphs refer to both Figures 4 and 5.

The Command Line Interface constitutes the physical layer interface to the IS-105 control bus. It uses a straightforward differential serial line to carry control messages between devices, and a differential driver and receiver circuit are needed to support this interface.

The IS-105 Communications Processor is responsible for controlling this interface, to send and receive messages on the bus. This function requires a mid-range to high-end 8-bit microcontroller, and some non-volatile RAM to store configurations in the event of a loss of power. The processor relays received messages to the Main Application Processor for disposition, and accepts messages from the main processor for transmission on the bus. Since the main processor is likely to require NVRAM as well, these requirements may be consolidated in the main processor's hardware. (There is no need to maintain separate NVRAM stores for both, since the processors will be tightly coupled.)

The Main Application Processor is probably a more capable device. maintaining larger buffers and therefore requiring more memory. and performing more complex tasks (such as supporting the User Interface at the TV), and thus perhaps needing a more powerful microcontroller. There are high-end 8-bit and 16-bit controllers which offer large RAM and ROM options. which may eliminate the need for external RAM and ROM. This processor also manages the interface to the Home Automation system. through the HA Interface Processor. The main processor is likely to require significant resources to manage the system-level issues associated with the HA interface.

The Video Generation circuitry is the hardware which the main processor uses to control the TV screen. in order to present menus and other information to the user. This circuitry will vary in implementation. depending upon the level of quality desired in the on-screen display. and single chips are available which implement much of this "character generator" functionality. The video Buffer presents the video signal to the IS-105 bus. The buffer is controlled by the main processor such that it is tristate-able and can use one of two or more video lines on the bus. in order to share the bus with another simultaneous session.

Finally, the device must contain a power supply. and since the box is not completely insulated (because of the connections to the other devices on the bus), the power supply is required to be line-isolated for safety reasons.

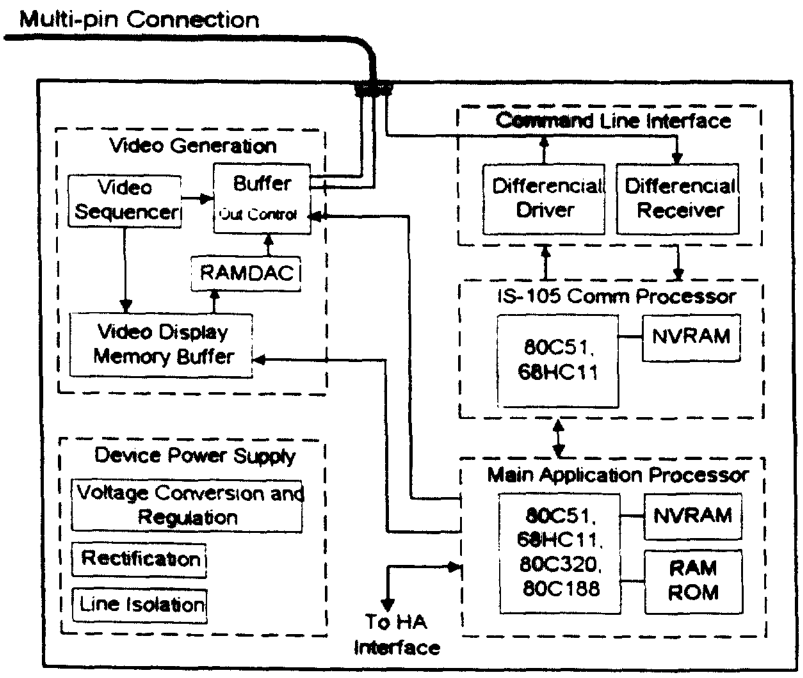

Figure 5. Common Elements in Home Automation System Interface (no audio)

The architecture described above is a straightforward approach which separates the real-time demands of the IS-105 bus from the application processor, by use of the IS-105 communications processor. It is certainly possible that the functions of both of these processors could be serviced by one microcontroller with sufficient processing capacity. Figure 6 presents a diagram illustrating this approach.

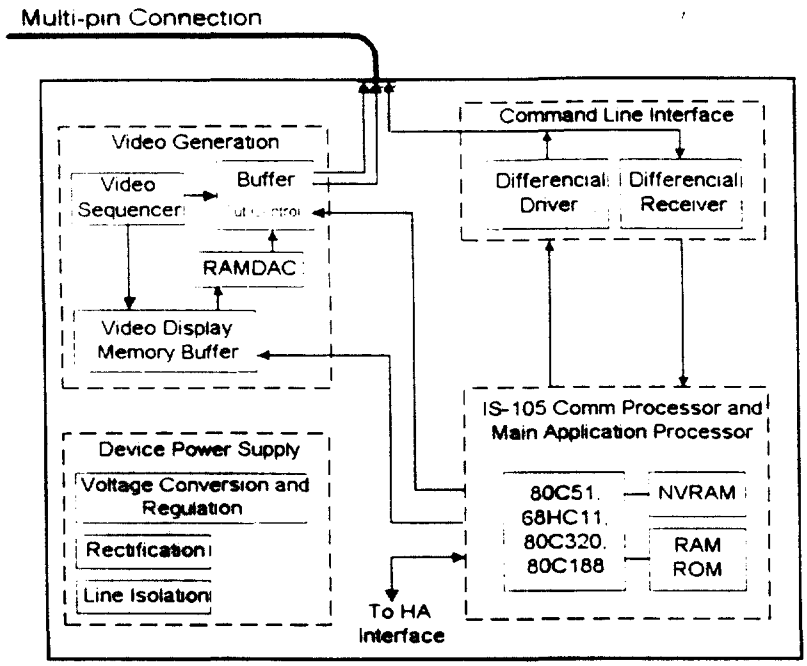

Figure 6. Common Elements, HA System Interface, Main and 105 Processors Combined

The features discussed thus far will be common to any implementation supporting the functionality outlined, independent of the HA system involved. This is an important point in the evaluation of the competitive effects of the proposed standard on various HA systems vendors—they will all share this base of hardware as a common requirement.

The selection of a particular HA system interface will determine the details of the HA protocol processor and associated PL interface circuitry, and will also affect the code in the main processor. The extent of this effect on the main processor is determined by the general requirements of that HA system. Other factors affecting this processor include the features which are to be supported by the interface device. Since the TV is such a convenient and capable display device, examples of such features include almost any feature which an installation and configuration tool might provide, as well as features associated with central controller devices in HA systems. Among these are: Automatic discovery of all devices in the home network; Assisted installation of new HA devices; Detailed reporting of device configurations; Diagnostic features to help in locating failed units; Recall and presentation of data collected by sensor devices; Direct manipulation of control values such as thermostat settings, lighting controls, sprinkler system controls, etc.; and Recall and manipulation of schedules for various automated subsystems. This list is only a sampling of the features which could be built into the HA Interface Decoder. The main processor would request/store and manipulate all of the data associated with these features, and present the formatted information on the screen.

Much of the resources required to support these User Interface features are comparable in different HA systems. It takes the same amount of memory to hold a Lighting schedule in one system as it does in the other, for example. And because the interactions with the User (mediated by IS-105 bus messages) are handled with HA language-independent tokens which simply report remote-control button-presses, the main processor will have an equivalent task in performing translations into necessary HA system messages regardless of the type of HA system.

We turn next to protocol-specific details of two example HA system interfaces: CEBus and LonWorks.

CEBus Interface Device Block Diagrams

Figure 7 depicts the overall HA interface device with a CEBus Power Line interface.

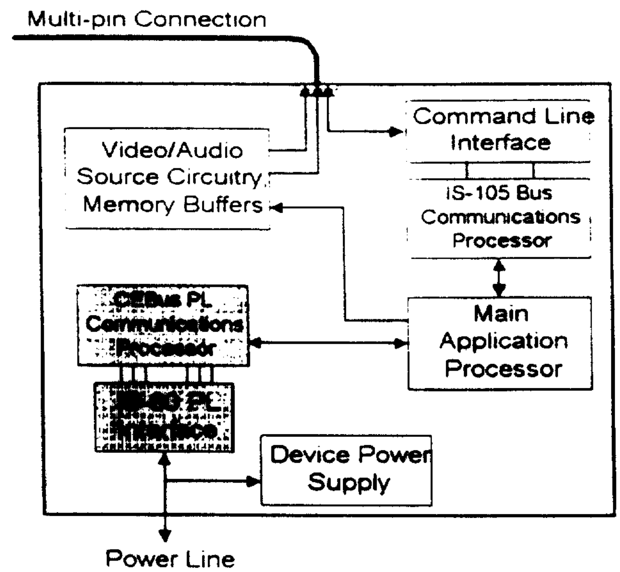

Figure 7. An IS-105 to CEBus PL Interface

The communications processor in the case of CEBus communications is likely to be an eight bit machine on the order of an 80C51 or 68HC11. Some non-volatile memory is required to store address assignments, device settings and User preferences. Depending on the implementation, the 8k to 12k of internal ROM of the mid-range versions of these chips may or may not be adequate for a complete solution. A high performance node also may require more than the 128/256 bytes of RAM available internally on mid-range 8-bit processors. High-end 8-bit microcontrollers meet these requirements.

Figure 8 presents one possible block diagram illustrating the circuitry associated with the CEBus Power Line interface. The circuitry is composed of a processor and its support circuitry, and the PL Medium Access Control (MAC) data pump controller, which deals with the transmission and reception of the spread spectrum symbols which comprise the PL messaging in IS-60.

The processor handles the higher levels of the Data Link Layer (DLL) CEBus protocol, performing error detection and correction, and duplicate packet rejection. Although it is conceptually necessary for this device to maintain non-volatile storage as shown in Figure 8, in reality the NVRAM of the main processor can provide this function.

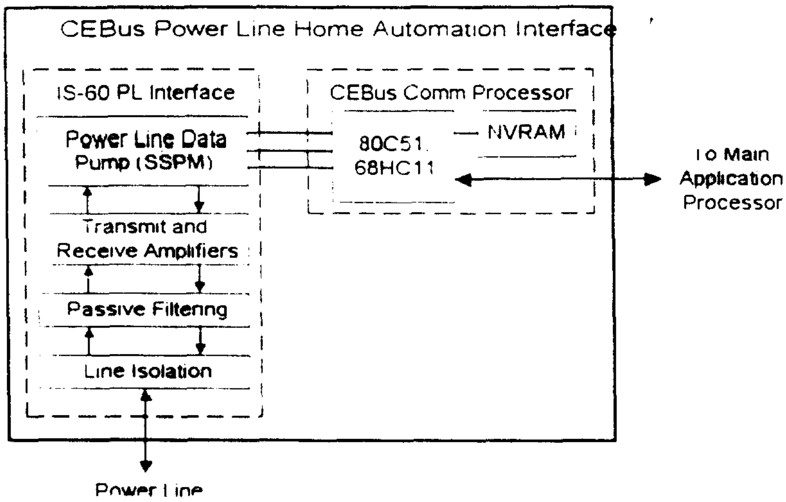

Figure 8. CEBus PL Home Automation Interface

LonWorks Interface Device Block Diagrams

Figure 9, below, depicts the equivalent interface device but with a LonWorks Power Line interface. The IS-105 Interface is unaffected by the selection of LonWorks as the main home automation bus standard.

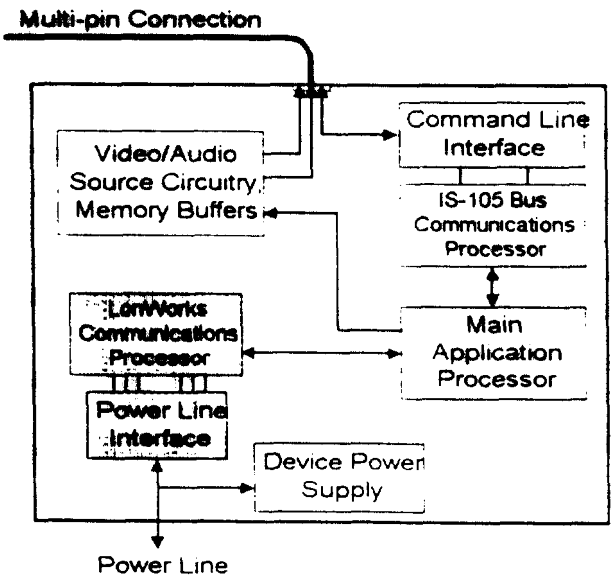

Figure 9. An IS-105 to LonWorks PL Interface

The hardware associated with the LonWorks HA system interface is considered next. Figure 10 presents one possible implementation. The communications processor in the case of LonWorks communications will be an Echelon Neuron. Since the main application processor handles all the heavy lifting the Neuron may require no external RAM or ROM, or an additional microcontroller.

It is possible that the best general design for the LonWorks version would include a Microprocessor Interface Program (MIP) enhancement to the main processor. so that the main processor would be more tightly coupled to the HA system. In this case the main processor would be unlikely to also handle the MAC and protocol layers of the IS-105 bus as shown in Figure 6. but instead would make use of a separate IS-105 processor as illustrated in Figure 5.

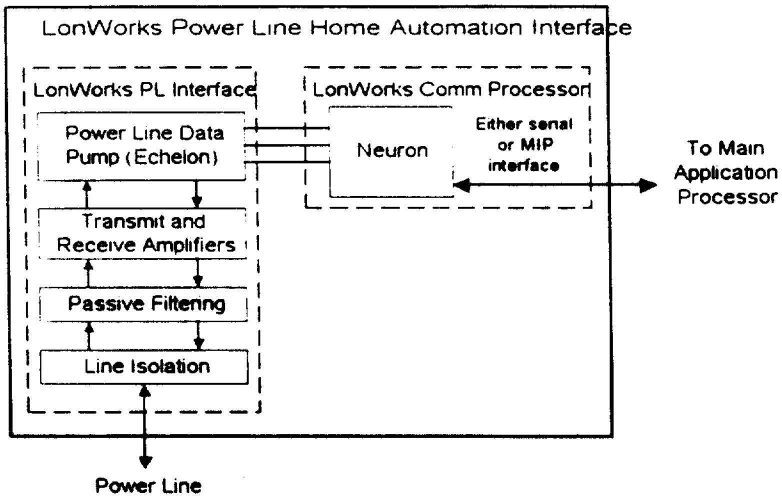

Figure 10. LonWorks PL Home Automation Interface

Conclusion

We believe that the cost of a practical implementation of a Home Automation interface to the proposed IS-105 bus will be substantially unaffected by the choice of the HA system. This conclusion is supported by the demonstration of the large proportion of circuitry which the various versions would have in common. as described in Figures 5 and 6. as well as by the substantial equality of the actual HA system interfaces required. as illustrated by specific example in Figures 8 and 10.

The IS-105 Decoder Interface is a reasonable and appropriate design. Although the use of a CEBus protocol subset on the bus may appear at first glance to favor this protocol over others, the truth is that the proposal favors no particular HA system.

References

1. EIA IS-105.1, Decoder Interface Standard. revision 5.1. with 4/18/95 mark-up
2. EIA IS-105.2. Decoder Interface Control Standard. revision 1.5, with 4/18/95 mark-up
3. Discussions with EIA staff regarding the recent addition of 4-bit-addressed. 8-bit tokens for general User Interface messaging (Markwalter: 6/2/95)